***Code:***

#include <iostream>

using namespace std;

void reflexive(int a[], int sizeOfA, int b[], int sizeOfB)

{

int i, j;

bool test;

bool hold = true;

int num = 0 ;

for(i = 0; i < sizeOfA; i++)

{

if(hold == true)

{

for(j = 0; j < sizeOfB;j++)

{

if(a[i] == b[j])

{

if(b[j] == b[j+1] || b[j]==b[j-1]){

num++;

hold=true;

break;

}

}

}

if(num==1){

num=0;

}else{

hold = false;

cout << "Reflexive - No" << endl;

return;

}

}

}

if(hold == true)

{

cout << "Reflextive - Yes" << endl;

}

}

bool Found(int a[], int sizeOfA, int b[], int sizeOfB){

bool ans = true;

for(int i=0;i<sizeOfB;i=i+2){

if(b[i] != b[i+1]){

ans = false;

break;

}

}

return ans;

}

void symmetric(int a[], int sizeOfA, int b[], int sizeOfB)

{

int i, j;

bool test;

bool hold = true;

if(sizeOfA==0 && sizeOfA==0){

cout << "Symmetric - Yes" << endl;

return;

}

if(Found(a,sizeOfA,b,sizeOfB)){

cout << "Symmetric - Yes" << endl;

return;

}

for(i = 0; i < sizeOfA; i++)

{

if(hold == true)

{

for(j = 0; j < sizeOfB;j =j+2)

{

if(b[j] != b[j+1])

{

int num1=b[j+2];

int num2=b[j+3];

if(b[j]==num2 && b[j+1]==num1)

{

hold = true;

}

else

{

cout << "Symmetric - no" << endl;

return;

}

j=j+2;

}

}

}

}

if(hold == true)

{

test = true;

cout << "Symmetric - Yes" << endl;

}

}

void antiSymmetric(int a[], int sizeOfA, int b[], int sizeOfB)

{

int i, j;

bool test;

bool hold = true;

if(sizeOfA==0 && sizeOfA==0){

cout << "AntiSymmetric - Yes" << endl;

return;

}

\\\

cout << "AntiSymmetric - Yes" << endl;

return;

}

for(i = 0; i < sizeOfA; i++)

{

if(hold == true)

{

for(j = 0; j < sizeOfB;j =j+2)

{

if(b[j] != b[j+1]){

if(b[j]== b[j+3] && b[j+1] == b[j+2]){

if(b[j+2] == b[j+3])

{

hold = true;

}

else

{

cout << "AntiSymmetric - NO" << endl;

return;

}

}

}

j = j+2;

}

}

}

if(hold == true)

{

test = true;

cout << "Antisymmetric - Yes" << endl;

}

}

bool Ans(int set[],int size1,int rel[],int size2)

{

int num;

bool ans = true;

if(size1==0 && size2==0)

{

return ans;

}

for(int i=0;i<size2;i++)

{

num = 0;

for(int j=0;j<size1;j++)

{

if(rel[i]==set[j])

{

num++;

}

}

if(num==0)

{

ans = false;

return ans;

}

}

return ans;

}

int main(){

int set1[0] = { };

int rel1[2] = {'a','a'};

cout<<"Test case 1:"<<endl;

if(!Ans(set1, 0, rel1, 2))

{

cout<<"invalid relation"<<endl;

}

else

{

reflexive(set1, 0, rel1, 2);

symmetric(set1, 0, rel1, 2);

antiSymmetric(set1, 0, rel1, 2);

}

int set2[0] = { };

int rel2[0] = { };

cout<<"Test case 2:"<<endl;

if(!Ans(set2, 0, rel2, 0))

{

cout<<"invalid relation"<<endl;

}

else

{

reflexive(set2, 0, rel2, 0);

symmetric(set2, 0, rel2, 0);

antiSymmetric(set2, 0, rel2, 0);

}

int set3[6] = {1,2,3,4,5,6};

int rel3[14] = {1,1,2,2,3,2,3,3,4,4,5,5,6,6};

cout<<"Test case 3:"<<endl;

if(!Ans(set3, 6, rel3, 14))

{

cout<<"invalid relation"<<endl;

}

else

{

reflexive(set3, 6, rel3, 14);

symmetric(set3, 6, rel3, 14);

antiSymmetric(set3, 6, rel3, 14);

}

int set4[6] = {1,2,3,4,5,6};

int rel4[16] = {1,1,2,2,3,2,2,3,3,3,4,4,5,5,6,6};

cout<<"Test case 4:"<<endl;

if(!Ans(set4, 6, rel4, 16))

{

cout<<"invalid relation"<<endl;

}

else

{

reflexive(set4, 6, rel4, 16);

symmetric(set4, 6, rel4, 16);

antiSymmetric(set4, 6, rel4, 16);

}

int set5[6] = {1,2,3,4,5,6};

int rel5[4] = {1,1,2,2};

cout<<"Test case 5:"<<endl;

if(!Ans(set5, 6, rel5, 4))

{

cout<<"invalid relation"<<endl;

}

else

{

reflexive(set5, 6, rel5, 4);

symmetric(set5, 6, rel5, 4);

antiSymmetric(set5, 6, rel5, 4);

}

return 0;

}

***Output:***

